Amendments to the Claims

Kindly amend claims 1, 6, 10 & 15, and cancel claims 3, 5, 7, 8, 12, 14, 16 & 17 (without prejudice), as set forth below. All pending claims are reproduced below, with changes in the amended claims shown by underlining (for added matter) and strikethrough/double brackets (for deleted matter).

1. (Currently Amended) A method of transferring executable program code between computer processes, the method comprising:

providing by a sender computer process an object which comprises a hashtable, the hashtable having at least one set of elements, one element of the at least one set of elements comprising executable program code, the providing comprising:

creating an empty hashtable;

integrating executable program code into the hashtable; and

serializing the hashtable into a serialized data object for transport to [[the]] a receiver computer process;

wherein the executable program code comprises logic which employs as ~~data input~~ the only input the hashtable within which the program code logic resides; [[and]]

transferring the object from the sender computer process to [[a]] the receiver computer process, retrieving the executable program code from the hashtable at the receiver computer process, adding data to the hashtable by the receiver computer process, and invoking the executable program code with the hashtable as the only data input thereto, wherein data is added to the hashtable by the receiver computer process prior to invoking of the executable program code retrieved from the hashtable with the hashtable as the only data input thereto;

wherein the providing comprises providing multiple serialized objects, each serialized object having a different hashtable therein, and transporting a first serialized object from a first sender computer process to the receiver computer process and transporting a second serialized object from a second sender computer process to the receiver computer process, and deserializing the first serialized object and the second serialized object at the receiver computer process to obtain a first hashtable and a second hashtable; and

wherein the method further comprises creating at the receiver computer process an empty common hashtable, integrating the first hashtable from the first sender computer process and the second hashtable from the second sender computer process into the common hashtable, adding data to the common hashtable at the receiver computer process, the data being relevant to executable program code in the common hashtable integrated therein from the first hashtable or the second hashtable, and being added prior to invoking the executable program code using as data input only the common hashtable, and iterating through the common hashtable for executable program code to be invoked using the common hashtable as the only data input thereto.

2. (Original) The method of claim 1, wherein the at least one set of elements comprises multiple tuples, each tuple comprising a first element and a second element, and wherein the second element of at least one tuple comprises the program code, and the second element of at least one other tuple comprises data relevant to the program code.

3. (Canceled).

4. (Previously Canceled).

5. (Canceled).

6. (Currently Amended) The method of claim [[5]] 1, wherein the first sender computer process, the second sender computer process, and the receiver computer process are on different computing units.

7. (Canceled).

8. (Canceled).

9. (Previously Canceled).

10. (Currently Amended) A system for transferring executable program code between computer processes, said system comprising:

means for providing by a sender computer processors an object which comprises a hashtable, the hashtable having at least one set of elements, one element of the at least one set of elements comprising executable program code, the means for providing comprising means for:

creating an empty hashtable;

integrating executable program code into the hashtable; and

serializing the hashtable into a serialized data object for transport to [[the]] a receiver computer process;

wherein the executable program code comprises logic which employs as ~~data input~~ the only input the hashtable within which the program code logic resides; [[and]]

means for transferring the object from the sender computer process to [[a]] the receiver computer process, for retrieving the executable program code from the hashtable at the receiver computer process, for adding data to the hashtable by the receiver computer process, and for invoking the executable program code with the hashtable as the only data input thereto, wherein data is added to the hashtable by the receiver computer process prior to invoking of the executable program code retrieved from the hashtable with the hashtable as the only data input thereto;

wherein the means for providing comprises means for providing multiple serialized objects, each serialized object having a different hashtable therein, and for transporting a first serialized object from a first sender computer process to the receiver computer process and for transporting a second serialized object from a second sender computer process to the receiver computer process, and means for deserializing the first serialized object and the second serialized object at the receiver computer process to obtain a first hashtable and a second hashtable; and wherein the system further comprises means for creating at the receiver computer process an empty common hashtable, for integrating the first hashtable from the first sender computer process and the second hashtable from the second sender computer process into the common hashtable, for adding data to the common hashtable at the receiver computer process, the data being relevant to executable program code in the common hashtable integrated therein from the first hashtable or the second hashtable, and being added prior to invoking the executable program code using as data input only the common hashtable, and means for iterating through he common hashtable for executable program code to be invoked using the common hashtable as the only data input thereto.

11. (Original) The system of claim 10, wherein the at least one set of elements comprises multiple tuples, each tuple comprising a first element and a second element, and wherein the second element of at least one tuple comprises the program code, and the second element of at least one other tuple comprises data relevant to the program code.

12. (Canceled).

13. (Previously Canceled).

14. (Canceled).

15. (Currently Amended) The system of claim [[14]] 10, wherein the first sender computer process, the second sender computer process, and the receiver computer process are on different computing units.

16. (Canceled).

17. (Canceled).

18-20. (Previously Canceled).

* * * * * *